Welcome to Networks and Security (CITS3002)

This unit introduces students to the design and implementation of contemporary wired and wireless computer networks, the systems- and application-level software necessary to support their efficient operation, and the security and privacy factors introduced by networks and their applications.

Today it is far more likely that a computer is connected to a computer network than not. As computer networks become increasingly faster, more reliable, and more pervasive, the way in which we view computer systems and computing is rapidly changing. This unit takes a bottom-up approach to explaining how current networking technologies work and the security threats and challenges that their use introduces.

The unit is presented in two parts:

**Data Communications:**

- Starting with an explanation of how data is packaged on physical media, such as on copper or optical cables, we follow with an explanation of how errors are introduced and how they can be both detected and corrected.
- We introduce a series of increasingly reliable and efficient network protocols which provide guaranteed, reliable message delivery on error-prone network connections.
- We next introduce local-area Ethernet, wireless, and mobile networks, and examine the security implications of the use of shared-media networks.
- We next examine the subject of routing protocols which enable messages to be both correctly and efficiently delivered between computers not directly connected.

**Internetworking:**

- We next examine the motivation for and design of the Internet, and its most frequently used protocols and applications, examining the general design of the TCP/IP protocol suite, the impact of the lack of a fundamental security model, and some common vulnerabilities and defences in using its protocols.
- We examine the basic building blocks of cryptography, followed by examples of how and where these techniques are often employed to secure network protocols and network-based applications.
- We continue with a discussion of the design and implementation of client/server applications using the Berkeley sockets API, synchronous and asynchronous I/O, iterative and concurrent servers, and partially automated approaches to developing network applications.

CITS3002 Networks and Security, Week 1: Introduction to Networks and Security, p1, 28th February 2018.
Some basic networking definitions

- A computer network is an interconnected collection of autonomous computers.
- Computers are interconnected if they are capable of exchanging information. The connections can be over copper wire, radio frequencies, optical fibre, infra-red, satellites, microwaves, or a sequence of these.
- Computers are autonomous if there is not a permanent master/slave relationship between them. Hence, a mainframe computer and its traditional peripherals do not constitute a computer network, but a desktop computer able to interrogate a more modern printer does.

Physically, a network is the computers and physical media connecting them.

Logically (and more interesting and relevant, here), a network is the software which connects and secures the computers their data, and services.

This unit focuses on computer networking software and its support by operating systems and programming languages.
Why do users want computer networks?

- Users can access shared and distributed data, e.g., the WWW, e-Commerce, B2B, airline reservation systems, cryptocurrency mining(!), from a variety of locations, including mobile and wireless services.
- Users can have their own (limited) computer and access shared and possibly distant physical resources (such as a printer, optical disk writer, cloud-based applications and backup, NAS music server).
- Networks can provide fault tolerance; if some hardware fails or network connectivity is intermittent, then logical shadowing and delay-tolerant networking enables data recovery and continued performance (again, cloud-based services are good examples).
- Networks provide cost benefits; the cost of 20 relatively powerful computers (on desktops or in a rack), and a central Web, email and file server is far cheaper to purchase (though maybe not to maintain!) than a "mainframe" or multicomputer providing the equivalent computational power by itself.
- Permit centralized facilities and remote collaboration - user-administration and file-management, windowing systems, software development with continuous integration, and tasks to minimize administration of these services.

Can you add to this list?
Research interests and networking

The combination of some significant problems provides a very rich body of investigation, some deep theoretical problems, and some important research results passed on to industry.

- **Unreliable communication** - messages sent over physical media are often garbled or lost. Programming (sequential and parallel) assumes reliable communication and this must be provided. Alternatively, if we accept that errors do occur, can we still manage to communicate with data loss (consider Mars probes, mobile communication, satellite TV and digital telephony)?

- **The support and study of temporal and spatial decoupling** of work patterns and communities. Sociologists are interested in communication and observation at a distance.

- **Privacy and security** (they are not the same thing) - industry's desire for electronic commerce and invasive marketing have driven the fields of authentication (and anonymity?) and encryption, and governments' desire for surveillance has driven the fields of identity and communication hiding.

- **Parallel programming** - as most recently seen in Hadoop and Map/Reduce models of processing distributed, and "big-data". Unlike sequential programming, writing, modelling and imagining the execution of parallel programs is very difficult. Parallel "activities" which must be resolved in parallel programs include distributed resource sharing and deadlock detection.

Can you add to this list?

---

CITS3002 Networks and Security, Week 1: Introduction to Networks and Security, p4, 28th February 2018.
The Threats to our use of Networks - Threat Categories

What is the *threat category*?
- interruption to service
- interception of data
- modification of data
- fabrication of data

Threats may be *active*
- self-reproducing worms (working independently),
- a logic bomb or botnet (needs a host program),
- a lurking trojan horse, trapdoor, or botnet program,
- a virus embedded in an application program, or
- a rootkit embedded in an operating system and utilities.

Threats may be *passive*
- keyboard logging,
- EMF monitoring and bridging air-gap networks,
- wired or wireless network packet sniffing, or
- simple social engineering.
The Threats to our use of Networks - The Attack Profile

Attack tools are designed by "troubled geniuses" -

- they have a deep understanding of your operating system (and access to source files),
- they look for known vulnerabilities (e.g. buffer overflows, race conditions),
- attackers have lots of spare time, and
- the tools are well written, adaptable, and avoid countermeasures.

A wide variety of tools are available on the Internet, downloaded and deployed by bored people andscript kiddies.

Most attack tools come with very good documentation - cookbook attacks.
The Threats - Personal and Professional Motives

- attacks may be indiscriminate,
- attacks out of curiosity, vandalism, peer pressures,
- attacks out of fear, greed, malice,
- political attacks - hacktivism,
- industrial espionage, and
- electronic warfare and attacks on infrastructure.

The Threats - Technical Goals

- Intelligence gathering (corporate or just fun),
- Denial of Service (corporate or just fun),
- Reading of protected information,
- Modification of protected information,
- Execution of arbitrary commands, with bonus points for gaining privileges and ensuring future access.
What is Vulnerable?

- people (e.g. use of social engineering),
- practices (e.g. using out-of-the-box configurations),
- programs (e.g. buffer overflows), and
- protocols (e.g. IP spoofing, source-level routing).

Reasons for Vulnerabilities

- insufficient testing (of core states, boundary conditions, and exceptional cases, no crash testing),
- dependence on flawed resources (race conditions, shared libraries),
- trusting untrustworthy data (forged packets, bad input data),
- inappropriate use of external resources ("escaping" to other programs),
- incompatible or incomplete design specifications (different behaviours by similar operating system kernels, incompatible interfaces),
- the complexity of large applications.
- the complexity of multiple, interacting applications.
The Threats - Outcomes

- financial gain for the attacker,
- incur a cost to victim,
- to gain media exposure (who does the media help?),
- to serve a political agenda, and
- servicing fear, greed, or a desire for fame.

Countermeasures

- improved education (for both awareness and use),
- reduced system complexity,
- increased software testing,
- improved physical protection,
- improved authentication,
- improved authorization,
- increased use of biometrics,
- improved auditing (intrusion detection),
- a better understanding of encryption, and
- introducing liabilities of software vendors.
So isn’t encryption the answer to everything?

“Cryptography is the science of making the cost of improperly acquiring or altering data greater than the potential value gained. The value of information usually drops with time, and cryptography makes the time required to obtain data in unauthorized ways long enough to decrease its value well below the money spent on obtaining it.”
—Jalah Feghhi, Digital Certificates

So what does it mean to trust a computer system?

“[Trust is] the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to control or monitor [that] party.”
—R.C. Mayer, An integrative model of organizational trust

“Trust in a passionate entity (human) is the belief that it will behave without malicious intent and trust in a rational entity (system) is the belief that it will resist malicious manipulation.”
—Josang (1996)

“Trust cannot be treated as a property of trusted systems but rather it is an assessment based on experience that is shared through networks of people.”
—Peter Denning (1993)
Are Open Source alternatives more secure?

There is no certainty to the claim, but the following observations suggest why there have been fewer (reported) attacks seen in the Open Source environment:

- The Open Source environments (including Linux and Apple's OS-X) have considerably less visible market share, making them less "interesting" (less of a target) for attackers.
- Modern Open Source applications (even under the Gnome and KDE managers) have fewer inter-application dependencies, and hence perform fewer actions "behind the backs of the user". In particular, applications store preferences in per-application locations, rather than a shared location (Registry).
- The choice of available applications is considerably more fragmented under Linux than under Windows (e.g. compare the high uptake of Outlook under Windows with the variety of Linux mail agents).
- The average Open Source user has a higher working knowledge of their platform, can anticipate a wider range of problems, and diagnose a wider range of symptoms.
- And most importantly - very few applications, such as mail clients and web browsers, support the immediate execution of attachments/ executables, let alone without first asking.

But the growth and reliance on The Internet of Things is about to rectify the imbalance!
The Need for Network Protocols

Definition: A computer protocol consists of an agreed format for messages, expressed by a packet header, an optional message component, and a set of rules for the exchange of messages between computers.

We see the use of protocols in Computer Science in almost every activity:
- World Wide Web servers communicate with Web clients/browsers such as Firefox, Chrome, and Edge using the Hyper Text Transfer Protocol (HTTP).
- Electronic mail and news articles are delivered and exchanged using the Simple Mail Transfer Protocol (SMTP) and the Network News Transfer Protocol (NNTP).
- Some operating systems (such as Linux) display their windows and graphics using the X-Windows Protocol or the impressive Virtual Network Computing (VNC) remote display system, and
- Computers share their local disks using the Network File Systems (NFS), samba, or the Windows-NT File System (NTFS) protocol.

Most importantly the protocol messages must:
- Happen in an agreed to order,
- Travel from the sender to the correct receiver, and
- Contain the correct, unambiguous, data.
The ISO/OSI Reference Model

With computer networks we require protocols to "meet" new computers, ask for information, agree to share data, etc.

The complexity of protocols can be simplified by separating some of the functions required into different protocols, and isolating "layers of responsibility" into different protocol modules.

e.g. at the lowest level we are concerned with correctly transmitting bits (0's and 1's) of data.

At another level we may be interested in transferring files between different computers which support different data type representations.

One solution to such separation of responsibilities is provided by the ISO (International Standards Organization) OSI (Open Systems Interconnection) reference model.
(This model was finally agreed upon and standardized in 1983).

Until about fifteen years ago, the OSI protocol suite was still actively sought in all tender specifications by the Australian and U.S. governments.

Today, the Transmission Control Protocol/Internet Protocol (TCP/IP) suite meets nearly all of our networking needs. While TCP/IP employs a 4-layer model, in contrast to the 7-layer ISO/OSI model, many identical concepts may be observed.
The ISO/OSI Reference Model, continued

“While [the ISO/OSI] model may or may not be a good way to organize real, live computer networks, it makes an excellent framework for organizing a book about them.

— Andrew Tanenbaum, Computer Networks (2/e), 1988

The deployment and acceptance of networking standards can be difficult, as evidenced by the irony of this early marketing literature:

“IBM is the only company that's shipping end-user networking software that conforms to OSI standards. The OSI stamp is important because it assures corporate users that the networking software will easily connect to other vendors' systems and software.

— IBM marketing literature, 1987

CITS3002 Networks and Security, Week 1: Introduction to Networks and Security, p14, 28th February 2018.
Why a Layered Model?

Why seven layers, why not 3, why not 10? The following principles were followed:

1. A layer corresponds to a different level of abstraction.
2. Each layer provides a well defined, independent function.
3. Within each layer unique protocol standards should be enforceable.
4. There should be a minimum of traffic between layers/across interfaces.
5. The number of layers should be sufficiently large that distinct functions are in different layers and that there are not too many layers for the whole model to become unmanageable.
The ISO/OSI Reference Model - layer-upon-layer

1. The Physical Layer

is responsible for transmitting a (raw) bit stream over the physical communication medium. As such it is concerned with the electrical and mechanical interface between the data and the physical medium.

The physical layer presents a bit stream to the layer above.
2. The Data-Link Layer

takes the bit stream from the physical layer and constructs logical chunks of data termed frames.

The purpose of framing is to ensure the reliable transmission of information by performing limited error detection and recovery.
The ISO/OSI Reference Model, continued

3. The Network Layer

is responsible for providing the connection between "end systems" across a network. These connections might include multiple, intermediate links and are intended to be independent of the (sub)networks used to transmit the data.

Network layer functions include:

- routing: deciding how to transmit frames between source and destination using addresses.
- relaying: enables data transfer (transparently) across intermediate (sub)networks.
- flow control: matches traffic flow with the physical capacity of a transmission path.
- sequencing: control ordering of frames across a network.
4. The Transport Layer

provides a reliable end-to-end service independent of the network topology. This is achieved by splitting messages into network sized packets and joining them back together again at the other end.

The transport layer often supports multiplexing to optimize network cost (several transport connections mapped into a single network connection) or splitting to enhance services (single transport to multiple connections).
5. The Session Layer

is the first upper layer crucial to *internet*working and manages the dialogue between end systems. Typically the session layer provides:

- establishment and closing of connections.
- synchronization to allow checking and recovery of data.
- negotiation of full and half duplex communication.
The ISO/OSI Reference Model, continued

6. The Presentation Layer

provides a standard format for transferred information by overcoming compatability problems between systems using dissimilar data encoding rules and (possibly) different display (input and output) technologies.
The ISO/OSI Reference Model, continued

7. The Application Layer

provides the interface between the application processes. In particular, functions such as file transfer, remote job execution (remote procedure calls) and application independent virtual terminal support are provided.

In overview, the application layer provides transparency to the users, load balancing between machines, data bases (banks and airline reservations), and the prospect of distributed operating systems.